

## Actividades. Descarga de archivos de Internet

### Actividad 1. Descarga un archivo arbitrario de Internet

En esta actividad tan solo tendrás que ejecutar el código que se te proporciona y comprobar que todo funciona bien, de manera que el recurso al que accedes se descarga correctamente en tu ordenador

### Actividad 2. El programa Stock Quote

En esta actividad aprenderemos a crear un programa que pueda leer precios del mercado bursátil desde Internet. Existen muchos sitios que ofrecen este servicio, como el portal Yahoo!

Visita <http://finance.yahoo.com>, introduce un símbolo cualquiera, como MMI.V, y pulsa el botón Get Quotes (Obtener cotización). Verás la información de cotización de Motorola, Inc., como puedes apreciar en la figura siguiente:

<b>Messina Minerals, Inc. (MMI.V) - TSXV</b>			
<b>0.02</b> 0.00(0.00%) Jan 24, 10:38AM EST			
Prev Close:	<b>0.02</b>	Day's Range:	<b>N/A - N/A</b>
Open:	<b>N/A</b>	52wk Range:	<b>0.01 - 0.07</b>
Bid:	<b>0.02</b>	Volume:	<b>0</b>
Ask:	<b>0.02</b>	Avg Vol (3m):	<b>67,919</b>
1y Target Est:	<b>36.70</b>	Market Cap:	<b>943.72K</b>
Beta:	<b>2.072</b>	P/E (ttm):	<b>N/A</b>
Next Earnings Date:	<b>N/A</b>	EPS (ttm):	<b>N/A</b>
		Div & Yield:	<b>N/A (N/A)</b>

**Figura 1. Cotización bursátil de Messina Minerals**

Lo que más nos interesa es la información mostrada en el navegador web

<http://finance.yahoo.com/q?s=MMI.V>

Haz clic con el botón derecho del ratón sobre esta página web y selecciona `Ver Código Fuente`, en el menú contextual para ver el contenido HTML; verás multitud de etiquetas HTML, con la información sobre MMI oculta. Modifica la línea de la clase `webSiteReader` del listado 1 para que imprima el contenido de esta página en la consola del sistema.

```
url = new URL("http://finance.yahoo.com/q?s=MMI")
```

También puedes almacenar toda la página en una variable `String` de Java en lugar de imprimir las líneas; basta con modificar el bucle `while` del listado 1.

```
String theWholePage;

String txt;

While(txt=buff.readLine() != null) {

    theWholePage=theWholePage + txt;

}
```

#### **Listado 1. Leer el contenido de index.html de google.com**

```
import java.net.*;

import java.io.*;

public class WebSiteReader {

    public static void main(String args[]){

        String nextLine;

        URL url = null;

        URLConnection urlConn = null;

        InputStreamReader inStream = null;

        BufferedReader buff = null;

        try{

            // index.html is a default URL's file name

            url = new URL("http://www.google.com" );

            urlConn = url.openConnection();

            inStream = new InputStreamReader(

                urlConn.getInputStream(), "UTF8");
```

```
buff = new BufferedReader(inStream);

// Read and print the lines from index.html

while (true){

    nextLine =buff.readLine();

    if (nextLine !=null){

        System.out.println(nextLine);

    }

    else{

        break;

    }

}

} catch(MalformedURLException e){

    System.out.println("Please check the URL:" +

                        e.toString() );

} catch(IOException e1){

    System.out.println("Can't read from the Internet: "+

                        e1.toString() );

} finally{

    if (inStream != null){

        try{

            inStream.close();

            buff.close();

        } catch(IOException e1){

            System.out.println("Can't close the streams: " + e1.getMessage());

        }

    }

}
```

```
}  
  
}  
  
}  
  
}
```

Si añades componentes (división en partes en función de los componentes del listado 2) a `theWholePage`, para deshacerte de las etiquetas HTML y de toda la información menos `Last Trade`, puedes crear tu propio programa de información bursátil.

Aunque este enfoque es muy útil para mejorar tus habilidades en el uso de componentes, puede que no sea la mejor solución, sobre todo si Yahoo! cambia las palabras que usa en su página. Por ello usaremos otra URL que muestra los valores bursátiles en formato CSV (valores separados por comas). La URL que debes usar para el símbolo MMI.V es la siguiente:

```
http://quote.yahoo.com/d/quotes.csv?s=MMI.V&f=s1ld1t1c1ohgv&e=.csv
```

Esta URL (URL del archivo de volcado de datos) genera una cadena que incluye el símbolo bursátil, la última cotización, la fecha y hora del precio, las ganancias por acción, el precio de apertura, la oscilación diaria y el volumen. Compara las líneas CSV con los datos mostrados en la figura 1, son iguales (en la imagen no se muestra la fecha).

```
"MMI", 7.81, "8/16/2010", "4:00PM", 0.17, 7.63, 7.86, 7.58, 25934265
```

Ahora, la tarea de división en componentes de la página web se reduce al análisis de una breve línea CSV. La clase `StockQuote` del listado 2 se encarga de ello: acepta el símbolo bursátil de la línea de comandos, obtiene los datos de Yahoo!, divide en componentes la línea CSV recibida e imprime el precio en la consola.

## Listado 2. Imprimir valores bursátiles.

```
import java.net.*;  
  
import java.io.*;  
  
import java.util.StringTokenizer;
```

```
public class StockQuote {

    static void printStockQuote(String symbol){

        String csvString;

        URL url = null;

        URLConnection urlConn = null;

        InputStreamReader inStream = null;

        BufferedReader buff = null;


        try{

            url = new

                URL("http://quote.yahoo.com/d/quotes.csv?s="

                    + symbol + "&f=sl1d1t1c1ohgv&e=.csv" );

            urlConn = url.openConnection();

            inStream = new

                InputStreamReader(urlConn.getInputStream());

            buff = new BufferedReader(inStream);


            // get the quote as a csv string

            csvString =buff.readLine();


            // parse the csv string

            StringTokenizer tokenizer = new

                StringTokenizer(csvString, ",");
```

```

String ticker = tokenizer.nextToken();

String price = tokenizer.nextToken();

String tradeDate = tokenizer.nextToken();

String tradeTime = tokenizer.nextToken();


System.out.println("Symbol: " + ticker +
    " Price: " + price + " Date: " + tradeDate
    + " Time: " + tradeTime);
} catch(MalformedURLException e){

    System.out.println("Please check the spelling of "
        + "the URL: " + e.toString() );
} catch(IOException e1){

    System.out.println("Can't read from the Internet: " +
        e1.toString() );

}

finally{

    try{

        inStream.close();

        buff.close();

    }catch(Exception e){

        System.out.println("StockQuote: can't close streams" + e.getMessage());

    }

}

}

}

public static void main(String args[]){

```

```
if (args.length==0){  
    System.out.println("Sample Usage: java StockQuote IBM");  
    System.exit(0);  
}  
  
printStockQuote(args[0]);  
}  
}
```